

## LISTING OF CLAIMS

### What is claimed is:

1. (Currently amended) A channel merging method for a VOD system, wherein said method comprising the steps of:

- (1) in response to requests from a plurality of clients for a video program, establishing a root channel (S1) and at least one sub-channel (S11), said root channel (S1) being established according to a request from a client that makes an earliest request, each of said sub-channels (S11) being established corresponding to a request of a client that makes a later request;
- (2) monitoring variation of a number of the clients that are using each established channel, and maintaining the established channel if the number of the clients using a monitored channel is not zero, and closing the established channel if the number of the clients using the monitored channel becomes zero,

wherein step (1) includes the sub-steps of:

- (1-1) searching a collection of root channels (S1) into which said sub-channels (S11) is possibly merged for a root channel (S1), said root channels (S1) satisfying the condition of  $\min(S11.start\_time - S1.start\_time) < object\_length/2$ , wherein the  $\min(S11.start\_time - S1.start\_time)$  indicates minimal values of the difference between the start time of said sub-channels (S11) and the start time of each root channel in said collection, and the  $object\_length/2$  indicates the half of the total length of the played video program;
- (1-2) if said root channel (S1) exists, searching a collection of posterity channels of said root channel (S1) for a parent channel (S6) into which said sub-channel (S11) will be merged, said parent channel satisfying the condition of  $\min(S11.start\_time - S6.start\_time) < S6.start\_time - S5.start\_time$ , wherein  $S6.start\_time$  indicates the start time of said parent channel (S6), and  $S5.start\_time$

indicates the start time of a parent channel (S5) of said parent channel (S6).

2. (Original) The channel merging method according to claim 1, wherein said root channel (S1) and each of said sub-channels are established in response to one of: a play starting request, and a program jumping request from a client.
3. (Original) The channel merging method according to claim 1, wherein said root channel and said sub-channels form a tree structure.
4. (Original) The channel merging method according to claim 1, wherein said step (2) includes the sub-steps of:
  - (2-1) indicating the number of clients that are using each channel as a count parameter;
  - (2-2) decreasing the value of said count parameter in response to occurrence of the event of merging, jumping or stopping of said each channel and sub-channels thereof;
  - (2-3) closing said channel on the server side if the value of said count parameter becomes zero.
5. (Original) The channel merging method according to claim 4, wherein said channel is maintained on the server side if the value of said count parameter is not zero, and said clients having performed the event of merging, jumping or stopping no longer receive the programs being played on said channel.
6. (Canceled)
7. (Currently amended) The channel merging method according to claim 1 ~~claim 6~~, wherein if said root channel (S1) is not found in step (1-1), said sub-channel (S11) is taken as a new root channel, its root channel parameter is set to be 1 and said sub-channel (S11) is the only channel being watched by said client.

8. (Currently amended) The channel merging method according to claim 1 ~~claim 6~~, wherein if said parent channel is found in step (1-2), said client watches the video program both on said sub-channel (S11) and said parent channel.

9. (Currently amended) The channel merging method according to claim 1 ~~claim 6~~, wherein if said parent channel is not found in step (1-2), the found root channel is taken as the parent channel of said sub-channel (S11), and said client watches the video program both on said sub-channel (S11) and said root channel.

10. (Original) The channel merging method according to claim 2, wherein if said client's request at time t is a starting request, a start time parameter is set to be t and an object offset parameter is set to be 0 in the sub-channel (S11).

11. (Original) The channel merging method according to claim 2, wherein if said client's request at time t is a jumping request, and the object offset time of said jumping is s, a start time parameter is set to be t and an object offset parameter is set to be s in the sub-channel (S11), and a stopping operation is performed on the channel on which said video program is played at time t and has been received by the client.

12. (Original) The channel merging method according to claim 5, wherein if said stopping operation is due to the ending of said video program, said sub-channel is directly closed and all the resources of said sub-channel are released.

13. (Currently amended) The channel merging method according to claim 1 ~~claim 6~~, wherein when secondary sub-channels of said sub-channel (S11) are established, steps (1-1) and (1-2) are repeated to take said sub-channel (S11) as the parent channel of the secondary sub-channels, and the value of the count parameter of said sub-channel (S11) is increased by 1.

**Serial No.: 10/726,835**

14. - 20. (Canceled)